

Домашнее задание 9. Шилова Ксения

Отчет по ошибкам

1.

```
== Java Exception: java.util.NoSuchElementException  
    at java.base/java.util.ArrayDeque.removeFirst(ArrayDeque.java:363)  
    at java.base/java.util.ArrayDeque.pop(ArrayDeque.java:594)  
    at com.company.Calc.calculate(Calc.java:164)  
    at com.company.Fuzzer.fuzzerTestOneInput(Fuzzer.java:9)  
DEDUP_TOKEN: ac725be893ee448b  
== libFuzzer crashing input ==  
MS: 0 ; base unit: 0000000000000000000000000000000000000000000000000000000
```

В строке 164 ошибка, возникает потому, что в классе Calc проверяется только случай, когда `stack.size() > 1`, а случай с нулем не учитывается. В этом тесте на вход подана пустая строка, значит, нужно исправить условие, чтобы оно проверяло неравенство нулю. Это можно было отследить с помощью `debug` класса `exception1`, который я прилагаю к работе.

Исправленный вариант:

2.

```

== Java Exception: java.lang.NumberFormatException: For input string: "zz"
   at java.base/jdk.internal.math.FloatingDecimal.readJavaFormatString(FloatingDecimal.java:2054)
   at java.base/jdk.internal.math.FloatingDecimal.parseDouble(FloatingDecimal.java:110)
   at java.base/java.lang.Double.parseDouble(Double.java:543)
   at com.company.Calc.calculate(Calc.java:152)
   at com.company.Fuzzer.fuzzerTestOneInput(Fuzzer.java:9)

```

Ошибка возникает в строке 152 класса Calc.java. В этой строке происходит Double.parseDouble.

Как видно из вывода `jazzzer`, строка “zz” не может преобразоваться в число типа `Double`. Поэтому, нужно пробрасывать `CastException` в такой ситуации:

Исправленный вариант:

3.

```
== Java Exception: java.lang.StringIndexOutOfBoundsException: String index out of range: 1
    at java.base/java.lang.StringLatin1.charAt(StringLatin1.java:47)
    at java.base/java.lang.String.charAt(String.java:693)
    at com.company.Calc.calculate(Calc.java:126)
    at com.company.Fuzzer.fuzzerTestOneInput(Fuzzer.java:9)
```

Ошибка в строке 126 класса Calc.java.

```
// Считываем первый символ
switch (sTmp.charAt(1)) {
    case '!':
```

Исправленный вариант:

```
// Считываем первый символ
switch (sTmp.charAt(0)) {
```

Первый символ нужно считывать с помощью индекса 0. В другом случае будет Exception в этом месте, что и показывает juzzer.

4.

```
== Java Exception: java.lang.UnsupportedOperationException: TODO: Не забыть реализовать оператор !
    at com.company.Calc.calculate(Calc.java:146)
    at com.company.Fuzzer.fuzzerTestOneInput(Fuzzer.java:9)
```

Ошибка вызвана исключение UnsupportedOperationException в строке 146 класса Calc.java.

```
case '!':
    throw new UnsupportedOperationException("TODO: Не забыть реализовать оператор !");
default:
```

Исправленный вариант:

```
if (stack.size() < 2 && sTmp.charAt(0) != '!') {
    throw new CalcException("Неверное количество данных в стеке для операции " + sTmp);
}
if (stack.size() != 1 && sTmp.charAt(0) == '!') {
    throw new CalcException("Неверное количество данных в стеке для операции " + sTmp);
}
```

Так как для факториала достаточно одного символа, то

```
dA = 0;
if (sTmp.charAt(0) != '!')
    dA = stack.pop();
```

...

```
case '!':
    if (dA < 0) {
        throw new CalcException("Неверный знак числа для этой операции");
    } else {
        for (int i = 1; i <= dA; i++) {
            dA *= i;
        }
    }
}
```

Затем программа работала без падений около 15 минут, мне показалось, что этого достаточно для того, чтобы сказать, что ошибок больше не обнаружится.